Review Problem 56

Is the following good Verilog? If not, fix it. If so, draw the circuit it represents.

\[
\begin{align*}
&\text{always}\_\text{comb begin} \\
&\quad \text{begin} \\
&\quad \quad c = \bar{d} \land \overline{p} \land \overline{c} \\
&\quad \quad p = \overline{c} \\
&\quad \text{el}se \\
&\quad \quad q = p \\
&\quad \text{end} \\
&\text{end} \\
\end{align*}
\]
Advanced Verilog - Complex "If" Synthesizer
<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>V</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

```
always @ (posedge ps) begin
    if (A == 0) begin
        if (B == 0) begin
            us <= 1;
        end
        else begin
            us <= 0;
        end
    end
end
```

```
// Next State Logic

logic [1:0] ps;
parameter [1:0] A = 2'b00, B = 2'b01, C = 2'b10;
```

**Advanced Verilog – "Case" Synthesis**
Advanced Verilog – "Posedge Statements"
end
else if (count) {
out <= 4'b0000;
}
always_ff @ (posedge CK) begin
logic [3:0] out;

Advanced Verilog - Counters
always @ (posedge clk) begin
  case (t
    2'b11: count11 <= count11 + 4'b0001;
    2'b10: count10 <= count10 + 4'b0001;
    2'b01: count01 <= count01 + 4'b0001;
    2'b00: count00 <= count00 + 4'b0001;
  endcase
endcase

Advanced Verilog - Histogram
Advanced Verilog – Less Than

end

lessThan = (a>b);

always_comb begin

logic [3:0] a, b;

draw diagram here
Advanced Verilog - Sequential Statements
Advanced Verilog – For Loops
SC or SD are closed
SA and SB are closed

When will current flow from the input to the output for each of these?

Review Problem 58
All circuit elements built from transistors

Readings: B.1-B.3.1

CMOS Transistors
Transistor Switches

However:

Switch open =

Switch closed =

Good 1

Good 1

However:

Switch closed =

Switch open =

False: 0 = 0 = 0 volts
True: 1 = 1.2 volts

That causes it to conduct.
A transistor is used.

P-Type

N-Type
Transmission Gate

$\theta = 0 = 1\ \Box$
$0 = 0 \Leftrightarrow \theta = 1 = 1\ \Box$

Make a switch that transmits 0 and 1.

Using Transistor Switches
How do we build a 2:1 Mux?

Multiplexors
Can have more than one source or signal

Bus, T, and Tresses:
Basic Gates

NOR gate:

Truth table:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Logic diagram:

- Inputs: A, B
- Output: C

Inverter:

- Input: A
- Output: A (inverted)

Ground (source of 0's):

- Input: A
- Output: 0

Power supply (source of 1's):

- Input: A
- Output: 1
A value can be read, but how written?

A pair of inverters can hold a value.

Memory